John von Neumann Institute for Computing

Boundary Element Method (BEM) with Parametric Surfaces

S. Grudinin, A. Baumgaertner

published in

From Computational Biophysics to Systems Biology (CBSB07),
Proceedings of the NIC Workshop 2007,
Ulrich H. E. Hansmann, Jan Meinke, Sandipan Mohanty,
Olav Zimmermann (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **36**, ISBN 978-3-9810843-2-0, pp. 129-132, 2007.

http://www.fz-juelich.de/nic-series/volume36

Boundary Element Method (BEM) with Parametric Surfaces

Sergei Grudinin[1,2] **and Artur Baumgaertner**[1]

[1] Institute of Solid State Research (IFF),
Research Centre Jülich, 52425 Jülich, Germany
E-mail: {S.Grudinin, A.Baumgaertner}@fz-juelich.de

[2] Institute of Neurosciences and Biophysics (INB-2),
Research Centre Jülich, 52425 Jülich, Germany

This work describes a new Boundary Element Method (BEM) implementation for biomolecular solvation with parametric surfaces. First, multi-scale volumetric synthetic electron density maps are constructed from parsed atomic location data of biomolecules using Gaussian isotropic kernels. Next, three different methods are used to extract triangular meshes for the molecular surface. They are: marching cubes, marching tetrahedra and marching tetrahedra with dual contouring. Then generated meshes are used in BEM electrostatic calculations. In this work we study: 1) calculation time and accuracy for multipole and direct electrostatic solvers; 2) energy convergence and calculation time with the density of boundary points different meshing algorithms; 3) energy convergence for different iterative linear solvers.

1 Introduction

Electrostatic interactions are known to play a key role in determining the structure and activity of biomolecules. Interactions with solvent are very important for biomolecular functioning. Here we present a new method for the implicit representation of solvent along with algorithms for calculation of electrostatic interactions.

2 Molecular Surface Representation

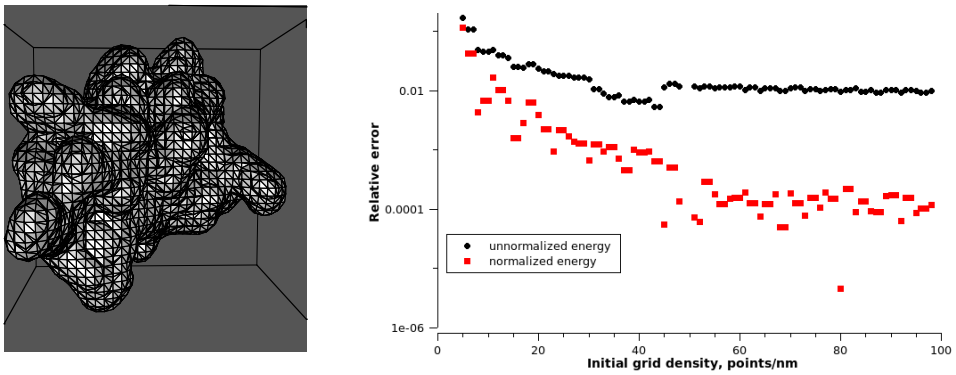
For our model we have chosen an implicit surface representation. The 3-dimensional scalar function $D(\vec{r})$ (Eq. 1), which is a sum of Gaussians centered at atomic nuclei, is similar to the electron density function. Then the isosurface $D = 0$ represents a molecular solvent-accessible surface with R_i parameters equal to WdV radii of corresponding atoms and a_i parameters responsible for surface smoothing[1,2].

$$D = \sum_i \exp^{-a_i(\frac{|\vec{r}-\vec{r_i}|}{R_i}-1)^2} \tag{1}$$

The Gauss map $\mathbf{S}$ of such a function is given in Eq. 2 and Gaussian K and mean H curvatures[6] can be calculated as shown in Eq. 3.

$$\mathbf{S} = \frac{1}{\|\nabla \mathrm{D}\|^3} \begin{pmatrix} d_{xx}(d_y^2+d_z^2) & -d_x d_y d_{yy} & -d_x d_z d_{zz} \\ -d_y d_x d_{xx} & d_{yy}(d_x^2+d_z^2) & -d_y d_z d_{zz} \\ -d_z d_x d_{xx} & -d_z d_y d_{yy} & d_{zz}(d_x^2+d_y^2) \end{pmatrix} \tag{2}$$

$$\begin{aligned} K &= det(\mathbf{S}) \\ H &= 1/2\, Tr(\mathbf{S}) \end{aligned} \tag{3}$$



(a) (b)

Figure 1. (a) Molecular surface of the BPTI protein. Marching cubes algorithm for the surface extraction has been used. (b) Error in solvation energy versus density of surface point for the marching cubes algorithm.

Marching algorithms have been chosen to extract the isovalue surface. We have used marching cubes and marching tetrahedra methods in our model. Difference between them will be discussed below. BPTI protein molecular surface, evaluated with marching cubes, is shown in Fig. 1(a).

3 Boundary Element Method

A well-known result for the induced charges σ on a surface between two dielectric media, which is a solution of Poisson's Equation, is given by:

$$\sigma = \left(\frac{\epsilon_{in} - \epsilon_{out}}{4\pi\epsilon_{in}}\right) \sum_i \frac{q_i(\mathbf{r} - \mathbf{r_i}) \cdot \mathbf{n}}{\epsilon_{in}|\mathbf{r} - \mathbf{r_i}|^3} + \left(\frac{\epsilon_{in} - \epsilon_{out}}{2\epsilon_{in}}\right) \sigma$$
$$+ \left(\frac{\epsilon_{in} - \epsilon_{out}}{4\pi\epsilon_{in}}\right) \oint \frac{(\mathbf{r} - \mathbf{r_s}) \cdot \mathbf{n}}{|\mathbf{r} - \mathbf{r_s}|^3} \sigma_s ds \qquad (4)$$

This equation can be solved in a matrix form $\mathbf{A}\sigma = \mathrm{b}$ using direct or iterative methods[4]. As the matrix size grows the iterative methods become more preferable. We have solved this equation with a number of modern iterative algorithms CGS, BiCGSTAB, GMRES, BiCG, QMR and Chebyshev iteration. All the algorithms are Krylov-subspace methods and provide well convergence. GMRES, unlike the other methods, needs only one matrix-vector multiplication per cycle and is a method of choice in many applications. QMR requires one matrix-vector and one transpose matrix-vector multiplications per cycle, but these two operations can be performed in a single function call in our modified DPMTA scheme[3]. So our implementation of QMR iteration is almost as fast as GMRES, but does not require a restart and uses less memory. All other algorithms, except Chebyshev, need two matrix-vector multiplications, which slow down their performance. Chebyshev iteration, although being very simple, needs some additional information about the spectrum of the matrix $\mathbf{A}$ and has been used here only for test purposes.

Since BiCG and QMR methods require a matrix transpose, we have extended DPMTA algorithm[3] to calculate the transpose of matrix $\mathbf{A}$. The multipole expansion $\mathbf{M}^{\mathrm{T}}_{\mathrm{n,m}}$ has a vector form and contains three new components:

$$\vec{\mathbf{M}}^{T}_{n,m}(\vec{r}) = \sum_{i=1}^{k} q_i s_i \vec{n}_i \mathbf{F}^{*}_{n,m}(\vec{r}_i) \tag{5}$$

Forces then calculated as:

$$q_i \nabla \Phi^{T}(\vec{r}) = Tr\left[q_i \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \vec{\mathbf{M}}^{T}_{n,m} \nabla \mathbf{G}_{n,m}(\vec{r}) \right] \tag{6}$$

Once the polarization charges σ are calculated, the solvation energy is evaluated as:

$$U = \frac{1}{\epsilon_{in}} \sum_{i} \frac{q_i}{|\mathbf{r} - \mathbf{r_i}|} + \sum_{k} \frac{\sigma_k}{|\mathbf{r} - \mathbf{r_k}|} \tag{7}$$

Finally the surface charges can be normalized according to Gauss's law:

$$\sum_{k} \sigma_k s_k = \frac{(\epsilon_{in} - \epsilon_{out})}{(\epsilon_{in} * \epsilon_{out})} \sum_{i} q_i \tag{8}$$

4 Code

The presented algorithm was written as a C/C++ extension to MMTK molecular modelling toolkit[5].

5 Numerical Tests

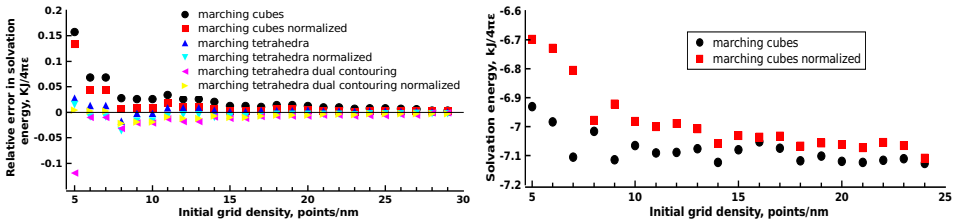
As a first example we have calculated solvation energies for a spherical cavity and compared them with analytical results given by Born equation. Results for this comparison are given in Fig. 1(b). Starting at point density $5\ points/nm$ the normalized energy error is below 1%.

Comparison for different iterative algorithms for the BPTI protein with 2500 surface elements is given in Tab. 1. GMRES and QMR methods perform the best, while Chebyshev did not converge due to a poor matrix $\mathbf{A}$ eigenvalue estimation.

	CGS	BiCGSTAB	GMRES	BiCG	QMR	Cheby
steps	24	23	30	36	37	-

Table 1. Convergence of iterative methods for the BPTI protein with point density 5 points/nm. Tolerance = 1e-6, 2500 surface elements, no preconditioner was used.

Numerical results for marching cubes, marching tetrahedra and marching tetrahedra dual contouring are given in Fig.2(a). All the methods show similar energy errors. Marching cubes provide a better mesh spacing and triangular quality, while marching tetrahedra



(a) (b)

Figure 2. (a) Three different marching schemes for a spherical cavity are compared. (b) Solvation energy for the BPTI protein at different grid point densities.

algorithm is simpler to implement. Convergence of the marching cubes scheme for the BPTI solvation energy is shown in Fig. 2(b).

6 Concluding Remarks

In the current work we have shown how to use a parametric solvation model together with BEM Poisson's solver. Three different marching schemes have been implemented and tested. We have also extended the DPMTA multipole algorithm[3] to deal with transpose matrices, which allowed us to use new iterative schemes[4]. The method has been coded as an extension to the MMTK molecular modeling toolkit[5] and will be soon available for download.

Acknowledgments

S.G. is thankful to Prof. Gerhard Gompper for the support in his postdoctoral studies.

References

1. The interpretation of protein structures: estimation of static accessibility. B Lee, FM Richards, J. Mol. Biol. **55**, 379-400, 1971.
2. Quality Meshing of Implicit Solvation Models of Biomolecular Structures. Y Zhang, G Xu and C Bajaj, Computer Aided Geometric Design **23**, 510-530, 2006.
3. DPMTA - A Distributed Implementation of the Parallel Multipole Tree Algorithm - Version 3.0. William T. Rankin `http://www.ee.duke.edu/Research/SciComp/Docs/Dpmta`.
4. A Historical Overview of Iterative Methods, D. M. Young, Comp. Phys. Comm. **53**, 1-17, 1989.
5. The molecular modeling toolkit: A new approach to molecular simulations. K. Hinsen, J. Comp. Chem **21(2)**, 79-85, 2000.
6. Modern Differential Geometry of Curves and Surfaces with Mathematica, A. Gray, 2nd ed. Boca Raton, FL: CRC Press, 1997.